Exploring Computational Thinking Through Collaborative Problem Solving and Audio Puzzles

Anna Jordan-Douglass
University of Wisconsin-Madison
Madison, WI USA
anna.jordandouglass@wisc.edu

Vishesh Kumar
University of Wisconsin-Madison
Madison, WI USA
vishesh.kumar@wisc.edu

Peter J. Woods
University of Wisconsin-Madison
Madison, WI USA
pwoods2@wisc.edu

IDC '18, June 19–22, 2018, Trondheim, Norway

ACM ISBN 978-1-4503-5152-2/18/06.
https://doi.org/10.1145/3202185.3210766

Abstract

Although educators, researchers, and designers have increasingly advocated for developing computational thinking (CT) in young children, the vast majority of CT learning environments fail to support the development of positive attitudes towards problem solving, confidence in dealing with complexity, and communicating and working with others to achieve a goal. To address this issue, our design team developed a music-based puzzle game called SynthSync. The game challenges players to work collaboratively to “debug” jumbled musical compositions through close listening, tinkering, and communication. SynthSync players manipulate controls to adjust musical variables (pitch, note length, and the length of rests) in arhythmic and dissonant musical puzzles based on popular songs until they “discover” the original piece of music.

Author Keywords

Computational thinking, collaborative problem solving, debugging, children, music, puzzles, game.

ACM Classification Keywords

J.5. Computer Applications: Arts and Humanities; Music

Introduction

Educational toys, software and curriculum are allowing for computational thinking and coding to be introduced to learners as young as 3 years old. Toys such as Cubetto and Think & Learn Code-a-Pillar are designed for 3-6 year olds to give sequential commands to trigger intended responses through the toy. Children 6 years old and up have access to many advanced robotic coding toys, and platforms such as Scratch Jr., Scratch, Hopscotch and Tynker to start creating with code. These resources work well for introducing programming concepts and providing a hands-on experience of creating computational artifacts, but creating with code without adequate scaffolding can lead to frustration and ultimately disengagement [4]. Engaging young learners in debugging practices before creating code is a productive way to grasp the fundamentals of programming, as well as foster interest in computational thinking practices while easing the challenges associated with making whole artifacts.

Making computational thinking (CT) concepts accessible to lower grade levels also requires fostering skills beyond understanding how a code block works or that the sequence of your blocks matters. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) [3] developed an operational definition of computational thinking as a problem-solving process that are supported by essential dispositions or attitudes that include:

- Confidence in dealing with complexity
- Persistence in working with difficult problems
- Tolerance for ambiguity
- The ability to deal with open ended problems
- The ***ability to communicate and work with others*** to achieve a common goal or solution

The need to develop these attitudes should compel educators and designers to look towards alternative environments that help students engage with the affective and emotional aspects of computational practices through easily accessible yet challenging tasks. Researchers should also consider the types of supports available to students in these environments. One way to develop these skills while simultaneously providing intrinsic support systems is through social collaborative engagement, achievable through collaborative problem solving. Collaborative problem solving (CPS) is composed of two main elements: the collaborative, sharing, or social aspects coupled with the knowledge or cognitive aspects [5]. By engaging players in CPS, they work together to achieve a group goal, each contributing to the solution, receiving feedback from the game to let them know when the group goal has been achieved. The game requires critical listening, plus communication, coordination, and cooperation.

With these considerations in mind, our design team has begun to develop a music-based puzzle game that leverages computational mechanics called SynthSync. SynthSync is an engaging, collaborative game that provides another space to practice and develop CT skills in a unique environment. These skills can then be applied to other problem-solving domains. We envision SynthSync as a tool to excite learners about CT through the process of solving tough audio puzzle problems and prepare them to move into creating computational artifacts with more confidence.

Sample Screenshots

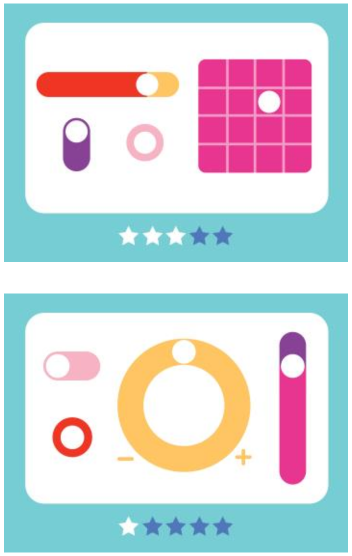

Figure 1. Sample screenshots of interfaces seen by players of SynthSync (alpha prototype). Different sets of controls are seen by different players (including dials, sliders, buttons, and XY-pads). All players also see success indicators at the bottom to have a visualization of their collective proximity to the final solution.

While the mechanics of SynthSync rely on the practice of debugging common to coding environments, the game embeds this mechanic within an audio environment precisely because of music's near universal ability to act as a social bonding technology [6]. Moreover, Bamberger [1] also notes that young students develop the foundation for musical intelligence simply through hearing and recognizing music that occurs in their natural environments. Music therefore lends itself to this research because it provides students with an intuitive learning environment through which they can engage complicated tasks and develop other complex thinking skills.

Design

SynthSync is a game for 6-10 year olds that aims to foster collaboration, listening, and tinkering through play with music. It allows players to collaboratively manipulate digital controllers that alter musical properties and solve interactive, adaptive audio puzzles. The game involves players controlling and modifying aspects of a musical piece being played, to figure out the connections between controls and music as players try to make the music "sound right." These musical pieces initially take the form of an audio puzzle, crafted from a familiar piece of music (such as Mary Had a Little Lamb). In their initial state as a puzzle, a selection of notes have been altered to make the piece sound unfamiliar. The pitches and lengths of some notes, along with the length between other notes, have been altered to create a dissonant and arrhythmic audio sample that only sounds vaguely similar to its original source material. Adding to this complexity, these alterations happen across instruments, leading to a jumbled mass of instruments playing against each other (as opposed to playing together). The task for the player is to untangle the dissonant elements and discover the original song.

In the game, the players work together to "discover" music by manipulating the controls to find the right pitch and length for a single note or group of notes as well as the length of time between notes. As the players collaborate to solve the audio puzzle, they are given visual feedback through an indicator bar that lights up as the team together finds the correct elements to solve the musical puzzle.

To play the game, three or more players work together collaboratively. Each player's screen has some set of controls - levers, knobs, or switches. When play begins, the players do not know how their particular controls will affect the puzzle. As they play, the players discover what impact each player's control has on the system. Through closely listening to the music, carefully decoding the relationship between the musical elements and the controls presented to the players, and collaborative problem solving, they will together solve the puzzles. The puzzle requires all players to participate to be solved.

The user interface relies on visual cues and audio feedback, keeping the experience free of text and accessible to pre- and early readers, and people with varying linguistic diversities. Each player's screen features different digital controls, such as sliders, knobs, or buttons, which each control different variables in the game (Figure 1). For example, if player one has a knob that adjusts "Variable A," which effects the first, fourth and eighth notes, they will rotate the knob until they find the right place to set it to lock those notes in place. The player knows when they have properly adjusted that control either by sound, if they recognize it, or by visual feedback from the indicator bar which will light up when the proper note has been identified. The player would then leave that knob alone and continue working with the rest of the team, engaging in collaborative debugging behaviors, until all variables have been found successfully, and the original song is discovered. This is the kind of task complexity known to be a strong predictor of productive team cohesion and collaboration [2]. As the players progress through the game, it gets more difficult by increasing the number of controls per player, as well as more variables controlled by player.

Future Work

Currently, our team is actively creating new puzzles to explore and utilize for playtesting purposes. We are also developing and testing feedback mechanisms to help players through frustrating moments, foster player communication and collaboration, and test visual design for the most effective visual puzzle feedback. We also plan to augment the engagement and embodied experience of modifying music by developing tangible interface controls with which children can play the game, as concrete physical manipulation, may support more effective learning [7]. While the current, digital state of our game gives us some understanding of the conceptual and theoretical practices at the foundation of the puzzles, the use of physical controls could have a significant impact on the kinds of collaborations and learning that occur when children engage with SynthSync. This exploration will happen tangentially to the development of new puzzles as well, resulting in a more robust learning environment for children to explore.

Eventually, we plan to extend SynthSync to include aspects of *creating* music with code, and one day envision an online community where makers can share, remix and critique one another's creations.

References

1. Bamberger, J. S. (1995). The mind behind the musical ear: How children develop musical intelligence. Harvard University Press.
2. Braarud, P. Ø. (2001). Subjective task complexity and subjective workload: Criterion validity for complex team tasks. International Journal of Cognitive Ergonomics, 5(3), 261-273.
3. Computer Science Teachers Association. "Operational definition of computational thinking." (2016).
4. Drosos, Ian, Philip J. Guo, and Chris Parnin. "HappyFace: Identifying and predicting frustrating obstacles for learning programming at scale." *Visual Languages and Human-Centric Computing (VL/HCC), 2017 IEEE Symposium on*. IEEE, 2017.
5. Fiore, Stephen M., et al. "Collaborative Problem Solving: Considerations for the National Assessment of Educational Progress." (2017).
6. Koelsch, Stefan. "From social contact to social cohesion—the 7 Cs." Music and Medicine 5.4 (2013): 204-209.
7. Marshall, Paul. "Do tangible interfaces enhance learning?." Proceedings of the 1st international conference on Tangible and embedded interaction. ACM, 2007.